Amendments to the Specification:

Please replace the paragraph on page 2, lines 9-18 with the following amended paragraph:

The inventors are aware of a multi-platform and language capable KB configurator referenced via an Information Disclosure Statement (IDS)[[IDS]] filed with this patent application. The system generates KB models reflecting user order parameters and can be accessed over a network by users through fat or thin clients operating a client plug-in module. Files are transferred over network lines as American Standard Code for Information Interchange (ASCII)[[ASCII]] files termed "flat files" in order to conserve bandwidth so light clients on such as hand-held devices plugged into the network can be used to configure and place orders. Further development and enhancement by the inventors has resulted in a capability for submitting changes or modifications to a KB configuration without requiring download of the KB or software to manipulate the KB. The technology uses one of a variety of universal object modeling languages like C++ or Java to implement the KB.

Please replace the paragraph on page 3, lines 1-5 with the following amended paragraph:

Although the inventor has contributed largely to the art, the systems described above are not capable of providing very complex rating information such as may be used to calculate complex pricing for services like medical ~~medial~~ insurance and other types of insurance coverage where final rates that must be calculated depend on accounting of many variables and client attributes and even state mandated rules.

Please replace the paragraph on page 4, lines 19-23 with the following amended paragraph:

In a preferred embodiment the Graphical User Interface is of the form of an interactive spreadsheet processing application and the computation model is a rating model. Also in a preferred embodiment the parser is adapted to read Extensible Markup Language (XML)[[XML]] and to write in Java Document Object Model structure. Also in a preferred embodiment the compilation component includes a lexical scanner and a code generator.

Please replace the paragraph on page 6, lines 3-9 with the following amended paragraph:

In yet another aspect of the invention a method for modeling a procedural rating schema is provided, comprising steps of (a) using a spreadsheet interface, inputting at least ~~last~~ one algorithm or formula into a specific cell or cells for rate calculation; (b) in the same interface, entering the required input values into cells and marking the cells that will carry the input values into the calculation; (c) in the same interface, marking the cell or cells that will show the output results from the calculation; and (d) saving the data as a rating computation file in Extensible Markup Language.

Please replace the paragraph on page 10, lines 11-18 with the following amended paragraph:

A persistence layer (PL) is provided as part of the foundation services for enabling applications automated and human-directed access to stored data for rate calculation, data mining, model building, data manipulation, and database searching of repositories 113. Objects are mapped to tuples stored in data repositories 113 using an Objected-Oriented Entity Relationship ~~OO/ER~~ mapping schema. A data workflow layer (WFL) is provided for the purpose of defining and maintaining modeled processes and workflow synergy between different users.

An application integration layer (AIL) is provided for enabling application-to-application integration and additionally to provide adapters to external systems.

Please replace the paragraph on page 15, lines 20-23 with the following amended paragraph:

A foundation of services (software layers) 201 is illustrated herein and is analogous to the software layers PSL, BLL, AIL, WFL, and PL described as foundation services software layers with respect to FIG. 1 above. These components will be formally ~~formerly~~ introduced herein with element numbers in addition to their labels of the example of FIG. 1.

Please replace the paragraph on page 19, lines 16-24 with the following amended paragraph:

FIG. 5 is a block diagram illustrating the function of DAAS 207 of FIG. 3 according to an embodiment of the present invention. DAAS ~~DASS~~ 207 enables automated access to data systems to requesting applications. DAAS 207 uses a service manager illustrated herein as service manager 500 ~~501~~ to identify data sources and their locations and to grant access to specific business objects 503 and specified attributes thereof. A registry 502 contains registered business objects that define the data contained in the databases and repositories. Shown here are databases 113 including a quote database, an enterprise database, a rate database, and a KBS. Commands 501 pass into the registry and are applied to or executed on the associated business objects to which the commands apply.

Please replace the paragraph on page 19, line 25 – page 20, line 7 with the following amended paragraph:

DAAS enables data creation, data reading, data updating, and data deleting. Configuration services and object locking capability are also DAAS-enabled ~~DASS-enabled~~ functions. Most data accesses by applications are entirely automated and performed in the context of processing an insurance quote or some other task ordered by a user. A platform service 504 having a proxy server component 505 makes the appropriate modifications to the respective data sources identified in the command request. Therefore, manipulations to a business object are translated into the appropriate manipulations to the actual data stored. For example, if a command results in a deletion of a specific attribute of a specific business object the platform service 504 validates the change and by proxy, effects the change in the object data. The next time that that particular business object is invoked, the attribute will not be there.

Please replace the paragraph on page 21, line 24 – page 22, line 6 with the following amended paragraph:

Once a user is authenticated both internally and by a third party, a role ~~roll~~ identification illustrated herein as role ~~roll~~ ID 703 is assigned to the user. The role ~~roll~~ ID determines what privileges the user will receive. For example, a privilege verification step 704 determines by assigned role ~~roll~~ ID whether the user will be allowed to view and or manipulate objects through DAAS ~~DASS~~ layer 304 and whether the user will receive access to workflow reporting through WFL 210. A policy governing privileges can be accessed from a policy store 705, which would also contain the role ~~roll~~ ID code or number assigned. Likewise any level of privilege at any level of granularity may be assigned a role ~~roll~~ ID so that if a user is verified to have the assigned

ID the user automatically gains the stated privileges for that ID. The users profile is of course also available and known to the system and may already have a role ~~roll~~ ID assigned.

Please replace the paragraph on page 26, lines 22-26 with the following amended paragraph:

Window 1202 has an input field "Alias" for naming the database. An input field 1204 is provided for specifying the driver of the database, which in this example is oracle.jdbc.driver.OracleDriver ~~oracle, jdbc.diver.OracleDriver~~. An input field 1205 is provided for specifying the location of the database. An input field 1206 is provided for specifying the name of the user accessing the database.

Please replace the paragraph on page 27, lines 17-19 with the following amended paragraph:

Cell B11 illustrates a calculated result value 354 and cell B13 ~~B12~~ illustrates a factor of 6.807692308. The calculated weekly rate is $2.86 as identified in cell B14. All calculated rate models are reviewable through interface 1400.

Please replace the paragraph on page 27, lines 20-26 with the following amended paragraph:

FIG. 15 is a screen shot of an interface 1500 revealing a first algorithm inserted as a function in cell B8. Window 1501 and 1504 are analogous to windows of the same description described with reference to FIG. 14. The displayed data, cells B5, B7, B11, B13, and B14 show the same data as the cells of FIG. 14 except ~~accept~~ that in cell B8, the inserted algorithm is

displayed instead of the result value 0.42. The algorithm is created using the "Insert" followed by "Function" as is practiced in other spreadsheet type applications. The algorithm is displayed as follows: